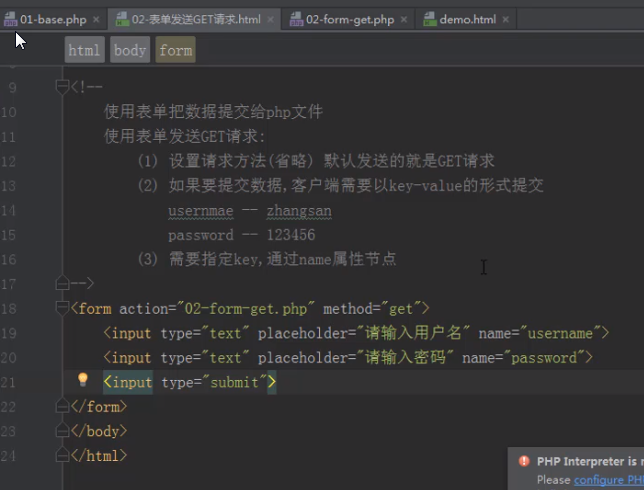
# Ajax发送网络请求

## Get发送请求

1. 前端最简单的表单发送用户密码请求，需要以key-value的形式提交数据，因此要设置name来设置key



2.后台PHP文件中判断username和password是否正确

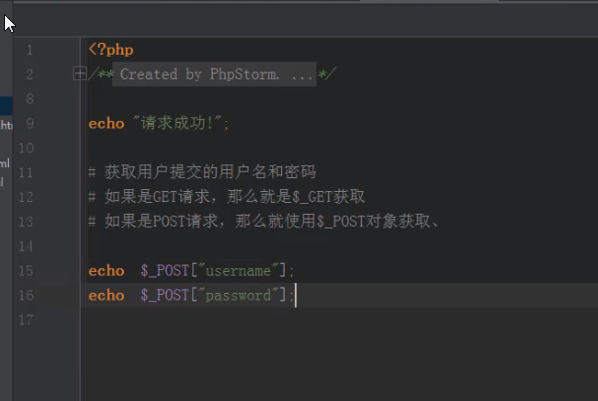


## Post发送请求

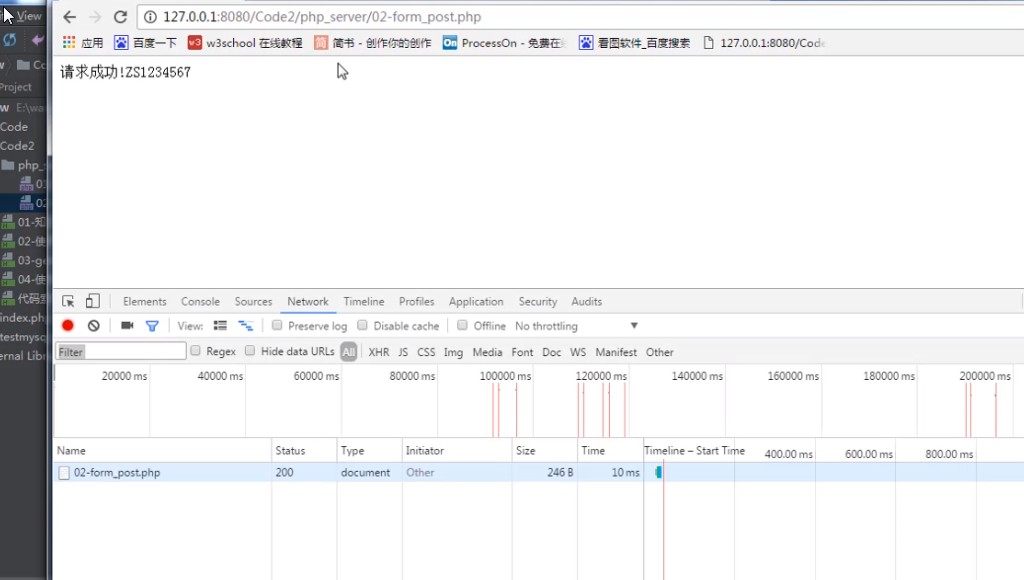
1.前端



1. 后台



页面输出结果



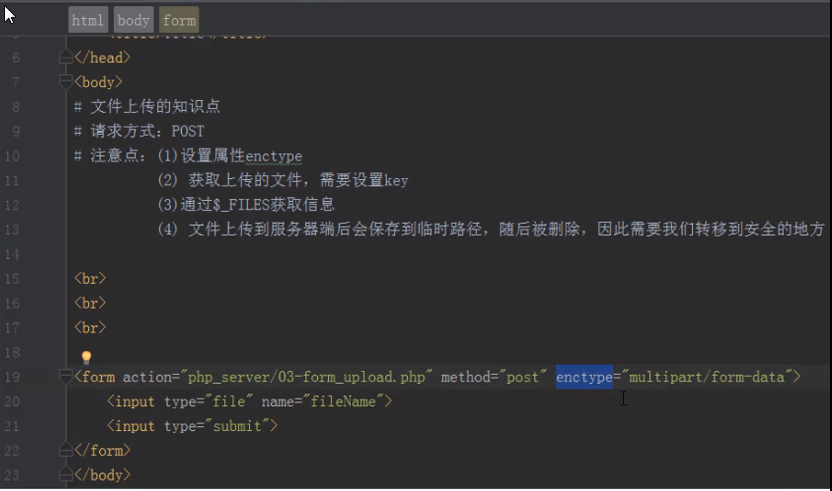
注册和登录可以用md5加密

# 使用表单作文件上传

（如果需要上传大文件，需要配置服务器环境）

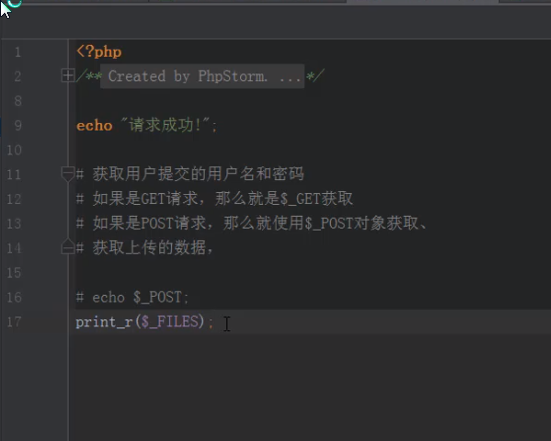
1. 文件上传资源比较大因此要用post方式上传
2. 文件上传要设置enctype属性，设置为multipart/form-data，意思是这是文件上传，请接收
3. 文件上传也是通过key-value的形式上传，因此需要设置name值来定义key，名字可以随便改
4. 文件上传后，后端php文件获取用户上传的信息用$\_FILES
5. 文件上传到服务器时会临时保存，随后被删除

前端：

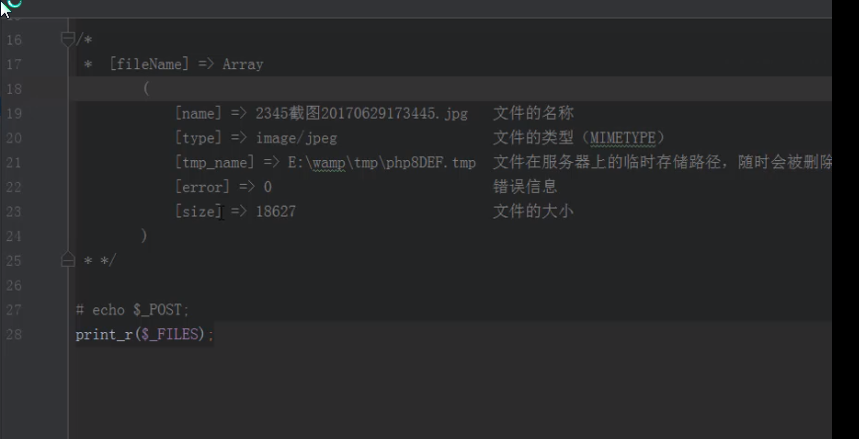


后端：

1. 用print\_r()来打印数据



2.打印出来的数据是数组，fileName为前端中设置的name（即key值）



3.由于上传的文件会临时存储后被删除，因此需要保存文件就要转移到安全位置，后端同move\_uploaded\_file()方法来转移文件。

注意：（1）在php语法中拼接url并不是用+，而是用.来拼接

